

```
1 <?php
2
3 /**
4  * @author GOLAY Brian
5  * @version 1.0 (2021/05/20)
6  * registration logic here
7  */
8
9 //require all the users functions
10 require_once("../sql/userDAO.php");
11
12 use FlightClub\sql\userDAO;
13
14 //check if the user is not already connected
15 if (isset($_SESSION['userID'])) {
16     header("Location: ../index.php?page=homepage");
17     exit();
18 }
19
20 /**
21  * Check the password strength with REGEX
22  *
23  * @param string $password
24  * @return bool
25  */
26 function password_strength($password)
27 {
28     //we check the length and initialize the default value for the return
29     $password_length = 8;
30     $returnVal = True;
31
32     //we check that the password is bigger than the password length
33     if (strlen($password) < $password_length) {
34         $returnVal = False;
35     }
36
37     //we check that there is a number in the password
38     if (!preg_match("#[0-9]+#", $password)) {
39         $returnVal = False;
40     }
41
42     //we check that there is at least one smaller character
43     if (!preg_match("#[a-z]+#", $password)) {
44         $returnVal = False;
45     }
46
47     //we check that there is at least one bigger character
48     if (!preg_match("#[A-Z]+#", $password)) {
49         $returnVal = False;
50     }
51
52     //we check that there is at least one special character
53     if (!preg_match("/[\\' ^$%&*()]{@#~?><>,|_+!-}/", $password)) {
54         $returnVal = False;
55     }
56
57     //we return the true if the password is ok or false if ko
58     return $returnVal;
59 }
60
```

```
61 //we filter the user input
62 $email = filter_input(INPUT_POST, "email", FILTER_SANITIZE_STRING);
63 $password = filter_input(INPUT_POST, "password", FILTER_SANITIZE_STRING);
64 $password2 = filter_input(INPUT_POST, "password2", FILTER_SANITIZE_STRING);
65 //we reset the message
66 $message = "";
67 //if all data are not empty
68 if (!empty($email) && !empty($password) && !empty($password2)) {
69     if (strlen($email) < 100) {
70         //we check that the email is not already in our database
71         if (userDAO::readEmail($email) == null) {
72             //we check booth of the password are the same and the strength of the
password
73             if ($password === $password2) {
74                 if (password_strength($password)) {
75                     //we hash the password
76                     $hashed = hash('sha512', $password);
77                     $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50,
70);
78                     $hashed = hash('sha512', $salted);
79                     //we hash again with password hash
80                     $hashed = password_hash($hashed, PASSWORD_BCRYPT);
81                     //we add it to the database
82                     userDAO::AddUsers($email, $hashed);
83                     $_SESSION['user'] = userDAO::readUsersByEmail($email);
84                     //we redirect him to the login page
85                     header("Location: ../index.php?page=login");
86                     exit();
87                 } else {
88                     $message = "Le mot de passe doit contenir au moins une majuscule,
une minuscule, un chiffre et un caractère spécial et faire 8 caractère ou plus!";
89                 }
90             } else {
91                 $message = "Les mots de passes ne sont pas identiques !";
92             }
93         } else {
94             $message = "L'email existe déjà";
95         }
96     } else {
97         $message = "L'email est trop long";
98     }
99 }
100
```